Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the applications:

Listing of Claims:

1. (Currently Amended) A circuit for performing n-bit cyclic redundancy check (CRC) calculations, comprising:

a plurality of CRC calculation blocks, each of the plurality of CRC calculation blocks performing a CRC calculation in parallel to yield a plurality of CRC calculation values of an n-bit CRC result;

a switch configured for selectively passing one CRC calculation value included in the plurality of ~~the~~ CRC calculation values, the passed CRC calculation value being calculated by one of said plurality of CRC calculation blocks; and

a CRC register for latching the one of the CRC calculation values selectively passed by the switch.

2. (Original) The circuit of Claim 1, wherein the latched CRC calculation value is inverted.

3. (Original) The circuit of Claim 1, wherein the latched CRC calculation value is fedback to the plurality of CRC calculation blocks.

4. (Original) The circuit of Claim 1, wherein the plurality of CRC calculation blocks receives a data input.

5. (Original) The circuit of Claim 4, wherein the data input is in the form of one or more eight bit bytes.

6. (Original) The circuit of Claim 5, wherein each of the plurality of CRC calculation blocks receives a different number of eight bit bytes from the data input.

7. (Original) The circuit of Claim 5, wherein each of the plurality of CRC calculation blocks use a 32 bit CRC polynomial.

8. (Original) The circuit of Claim 7, wherein each of the plurality of CRC calculation blocks uses the same 32 bit CRC polynomial.

9. (Original) The circuit of Claim 8, wherein the 32 bit CRC polynomial is for Ethernet.

10. (Original) The circuit of Claim 5, wherein there are sixteen CRC calculation blocks including the first and second CRC calculation blocks.

11. (Original) The circuit of Claim 1, wherein the circuit is implemented as a field programmable gate array.

12. (Original) The circuit of Claim 7, wherein the 32 bit polynomial is loadable into the circuit.

13. (Original) The circuit of Claim 7, wherein the 32 bit polynomial is built into the circuit.

14. (Currently Amended) A method for calculating a cyclic redundancy check (CRC) value with a variable width data input, comprising:

inputting a variable width data word;

calculating a first CRC value having a first number of bits using the variable width data word; ~~and~~

calculating a second CRC value having a second number of bits using the variable width data word, wherein the first and second CRC value calculations occur in parallel; and

selecting one of: the first CRC value and the second CRC value as the CRC output value

wherein the selection of one of: the first CRC value and the second CRC value is accomplished by a switch selection signal.

15. (Original) The method of Claim 14, wherein the variable width data word is 32 bits long.

16. (Original) The method of Claim 14, wherein the second CRC value is calculated using a portion of the variable width data word that is not used in the calculation of the first CRC value.

17. (Original) The method of Claim 16, wherein the second CRC value is calculated using a portion of the variable width data word that is used in the calculation of the first CRC value.

18. (Cancelled)

19. (Currently Amended) The method of Claim ~~18~~ 14, further comprising feeding back the CRC output value as an input for the calculation of the first and second CRC values.

20. (Cancelled)

21. (Currently Amended) The method of Claim ~~20~~ 14, wherein the switch selection signal is a multi-bit value.

22. (Original) The method of Claim 21, wherein the multi-bit value is decoded to provide a four bit value to a mux that performs the selection of the one of the first and second CRC values.

23. (Original) The method of Claim 22, wherein the CRC output value is latched after being output by the mux and the latched CRC output value is fedback.

24. (Original) The method of Claim 23, wherein the CRC output value is inverted.